Experimental analysis of the influence of context awareness on service discovery in PNs

Olsen, Rasmus Løvenstein; Nickelsen, Anders; Nielsen, Jimmy Jessen; Schwefel, Hans-Peter; Bauer, Martin

Published in:
Proceedings of IST Mobile and Wireless Summit, Mykonos

Publication date:
2006

Document Version
Publisher's PDF, also known as Version of record

Experimental analysis of the influence of Context Awareness on Service Discovery in PNs

Rasmus L. Olsen[1], Anders Nickelsen[1], Jimmy Nielsen[1], Hans-Peter Schwefel[1], Martin Bauer[2]
[1] Center for TeleInFrastruktur, Aalborg University, Niels Jernes vej 12, 9220 Aalborg, Denmark
Email:[1]{rlo|anic03|jjni03|hps|}@kom.aau.dk
[2] NEC Europe Ltd., Network Laboratories, Kurfürsten-Anlage 36, 69115 Heidelberg, Germany
Email: martin.bauer@netlab.nec.de

***Abstract*— In this paper we present an experimental prototype for context aware service discovery (CASD) specifically aimed at Personal Networks. The goal is to use context information to provide the user with those services that are especially useful in his/her current situation. We describe the concept of CASD, a two-level architecture and the necessary components for performing CASD in Personal Networks. We also provide experimental performance evaluation of CASD. These results are compared to service discovery mechanism without context awareness. We show that there is some overhead associated to the CASD process, in terms of response time, while the generated network traffic may be reduced due less service description needs to be transferred between nodes.**

***Index Terms*— Context aware service discovery (CASD), Personal Networks, experimental evaluation**

I. Motivation and background

Service discovery is an important technique for a user to locate services in a network. Without service discovery, a user must know about the presence of a service on a specific device in order to use it. In networks with a potentially large number of devices, service discovery can also help the user in managing available services on the network. This is particularly the case when considering Personal Network.

Personal Networks, which has been introduced in [1], consists of the user's mobile devices in his/her Personal Area Network (PAN), plus other personal devices, organized in subnetworks e.g. at home or the office. The different subnetworks are connected by some interconnecting structure. The concept of Personal Networks is illustrated in Figure 1.

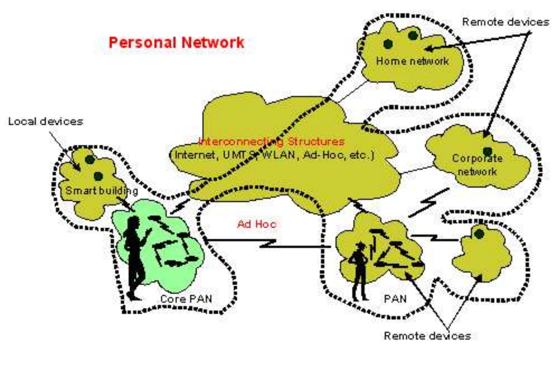

Fig. 1. Example illustration of the Personal Network concept, [1].

Personal Networks have been under investigation in the EU IST project "My Personal Adaptive Global Net (MAGNET)" [2] and are continuously being studied in "MAGNET Beyond"[3], with focus on federated PNs.

In PN's, there may be a large amount of devices, with each providing several services. Not all of these services are useful to the user at the same time. In certain situations it would be helpful if the service discovery system could filter services that are not relevant away. An example could be a case where a user requires a location specific service, for instance a printer. In a Personal Network searching for printers, a normal service discover will reveal all connected printers to the user. When taking into account the context of the user by using the geographical location, then only printers nearby the user will be shown to the user. Furthermore, if the printer queue on the nearest printer is rather large, it may be better to select a printer that is a bit further away. This is just one example of how two information, location and printer queue, can be used to filter away and/or order less relevant services. Other types of information can also be used to enhance service discovery, user preferences, ambient light and sound levels. Using such information to enhance service discovery we call *context aware* service discovery (CASD). Requirements and architecture for CASD has been studied in [4] and [5].

Similar work on context aware service discovery has been conducted in e.g. a system called Ad hoc Context Aware Network architecture (ACAN)[6], which is aimed at wireless settings, and not scalable to global level. However, most legacy service discovery systems, such as UPnP [7], SLP [8], Salutation [9], Jini [10] etc. only implements context awareness to some degree, and surely not to a sufficient level which is needed for Personal Networks.

In this paper we present an experimental performance evaluation of a CASD aimed at Personal Networks. The results are compared to service discovery which is not context aware. We show that there is some overhead associated to the CASD process, in terms of response time, while the generated network traffic may be reduced due less service description needs to be transferred between nodes.

II. Service discovery architecture

A. High level architecture

Since PN's may become rather large, the service discovery system must be able to cope with network sizes ranging from small local networks to globally scaled networks. For this reason, the architecture is split into two layers, one for local discovery which covers all nodes within a local network domain and one for global discovery which covers all interconnected network domains. The concept is illustrated in Figure 2. In this constellation two entities play an important

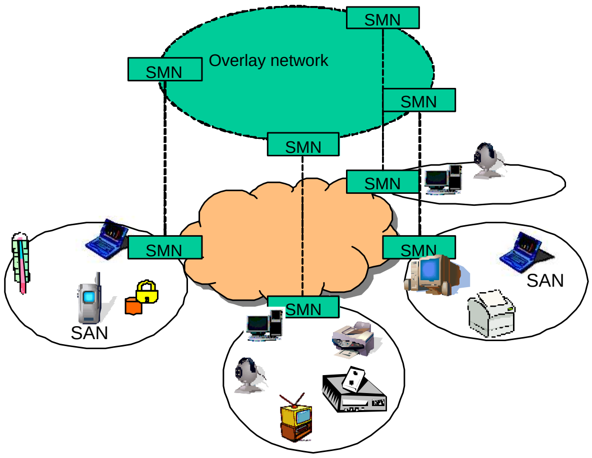

Fig. 2. MAGNET Service Discovery overlay network for service discovery, with an SMN in each cluster and SAN's in some.

role, the Service Management Node (SMN) and the Service Assistant Node (SAN). Their role and definitions are described in the two bullets.

- **Service Management Node** (SMN): A node that maintains a repository of context data within a cluster of nodes. This node is also responsible for the interaction between itself and other SMN's in the overlay network.
- **Service Assistant Node** (SAN): A node which can take over the role as an SMN, e.g. in cases where the SMN fails for some reason. It may also act as a proxy between the SMN and non IP devices.

B. Service Management Layer

We define a layer on top of the transport layer which we call the Service Management Layer, Figure 3. This layer constitutes the components that perform CASD. The modules

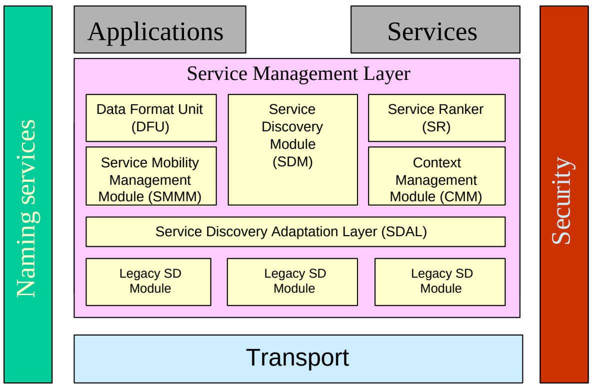

Fig. 3. Components in the Service Management Layer.

we base our measurements on are UPnP and INS/Twine as two of the legacy SD modules. The INS/Twine module is used to allow the SMN for to perform global service discovery, which is done in an overlay peer to peer approach. UPnP is used for local cluster level discovery. The Service Discovery Adaptation sub Layer (SDAL) provides a coherent interface between the upper and lower components. Furthermore, the SDAL offers access functionality to the SMN service repository. The Context Management Module uses the SDAL to discovery where to obtain context information, and to maintain the values dynamically. The Service Discovery Module (SDM) together with the Service Ranker (SR) are the key components in enabling CASD, since these are used to control and rank services in the discovery process. When a service has been discovered they are given a score value using the Service Ranking module, and hereafter ranked according to the level of interest of the user. Further details of the SML and the components can be found in [11].

III. Context Aware Service Discovery

In this section we describe in more details the activities between the components and devices when CASD is done. The description is divided into a local and a global part, to reflect the two layered service discovery approach.

A. Local discovery

The first part is the local discovery process, which is illustrated in Figure 4. Initially an SMN receives a service discovery

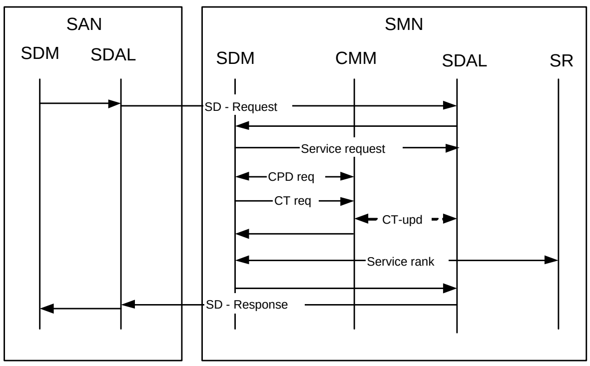

Fig. 4. High level message flow diagram for local CASD at the SMN

request from some user, which could be another software entity, application or from a SAN (as illustrated in Figure 4). This triggers the SDM to make a lookup in the local service repository maintained by the SDAL. If any services are found, it obtains a set of parameters, specific to that service, that will used in the ranking process from the CMM module. In this paper we use the term Context Parameter Data (CPD) for such information. How these are constructed and what they constitute will be described in more detail in Section III-C. It is the responsibility of the CMM to discover and maintain these parameters. In case they are not present locally, the CMM will try to discover these parameters. Next step in the process is for the SDM to obtain the value of relevant information, e.g. location of the user, which changes over time.

More details of this process can be found in [11]. The last part of the process is to put a score value on each service based on context information and the CPD's obtained. The ranking is then done by sorting the services according to their individual score values, which are then presented to the requester.

B. Global discovery

For the global part, the required communication can be seen in Figure 5. Initially, the overlay network is used to discover

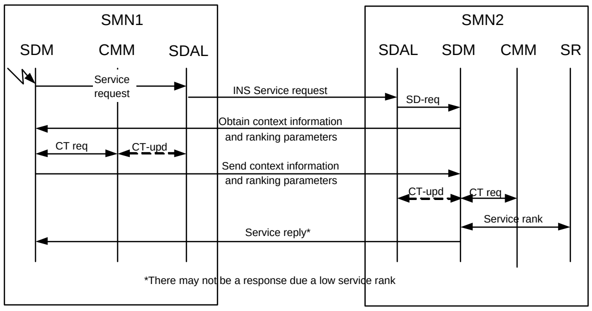

Fig. 5. High level message diagram of global CASD.

the SMNs which has services of the particular service type. Those SMNs that holds requested services, responds to the requesting SMN in order to obtain the CPD and relevant context information. In the figure this is SMN2. The SDM at SMN2 sends a request back to the SMN to obtain relevant CPD and context for SMN1. Meanwhile SMN2 requests its local CMM for context information at SMN2. When CPD, context for SMN1 and SMN2 is known, the score value is calculated. If this value is beyond a certain threshold, the service description is send to SMN1 with its score, where the response along other responses from other SMN's is ranked according to each individual score values. The final step is then to present the result to the user.

C. Context based service ranking

The heart of CASD is in the ranking module, i.e. when calculating a score value for services based on context information, and sorting the found services after their individual score. The basic concept of the service score calculation is given in Equation (1).

$$S_m = \frac{\sum_{n=1}^{N} w_n f_n(\Delta x_n)}{\sum_{n=1}^{N} w_n} \tag{1}$$

where S_m is the total score of service m, w_n is the nth weight, x_n is the nth context variable (Δx here represents the context difference between the desired and the actual context value) and $f_n(x)$ is a context score function. The functions, parameters and inputs varies from service to service, and is a part of the CPD mentioned earlier. For instance, a printer service should be ranked based on different context information than a gateway service. Hence, the ranking system must be flexible and extensible enough to cope with a high variety of services and context information.

1) Score function:

The following four functions are a part of our implementation:

- **Below**: A function that rewards values below certain values
- **Above**: A function that rewards values above certain values
- **Between**: A function that rewards values between a certain value range
- **Avoid**: A function that rewards values outside a certain value range

2) Weight parameter:

Different aspects of the context service discovery need to be taken into account when weighing the output scores. Potential information are:

- Context mismatch probability, i.e. the probability that some context value used in the ranking process has changed to another value.
- The accuracy of a measured context information: If a location is known to be 100m away with a precision of $\pm$ 50m, this may easily be worse than a service being 125m away $\pm$ 5m.

These aspects are important, and can be used to influence the ranking by weighing the output of the score function. This has not yet been implemented, but will be left for future study.

IV. PERFORMANCE MEASUREMENTS

For evaluation of the proposed CASD, we focus on the following performance parameters in this paper:

- Response time: The time from a user pressing the discovery button till the service has been found
- Traffic overhead: The amount of traffic generated
- Average Bandwidth: The average bandwidth usage during the discovery, i.e. the traffic overhead divided by the test time.

The test is carried out two times, one with context awareness and one without context awareness, in order to compare and evaluate the impact of the implemented context awareness.

A. Test bed and procedure

The experimental setup is seen in Figure 6, and consists of three computers: two are connected through a fixed network using a router, and the third having a wireless 802.11g connection to one of the laptops. We limit our evaluation to

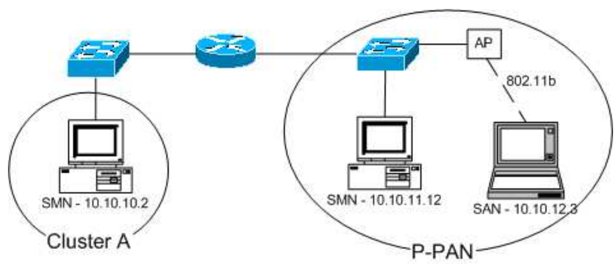

Fig. 6. Experimental test bed setup

three cases, in which services already have been registered at the SMN's.

1) Local discovery, when the user triggers a local service discovery from the SMN.
2) Local discovery, when a user triggers a local service discovery from a SAN.
3) Global discovery, when the user triggers a global PN service discovery from the SAN.

The discovery time will be measured at the node from where the user is triggering the system, i.e. for 1) this will be the SMN, for 2) and 3) this will be the SAN. Since the SMN will play the key role in the discovery, network traffic is measured to/from the SMN.

B. Input parameters

For all three cases we will investigate how the amount of registered services influences the performance parameters in the discovery process. The following parameters are kept fixed during the tests:

- Fixed number of UPnP devices: 4 on each node
- Fixed number of context dependency, i.e. CPD's per service: 4 per service
- We use a reactive context information update scheme
- The size of a device is given by 1,2 KB + 0,33 kb/service for normal, and 1,2kb + 0,89kb/service for CASD)
- User request inter arrival time is deterministically fixed to three seconds
- No other traffic is generated on the test bed, except what is created by the INS overlay network, i.e. between the two SMN's. This traffic is approximately 16kb/s.
- For each point, 30 measurements are taken and the mean with a 95% confidence interval is calculated
- The two SMN's are 3GHz Pentium machines with each 256MB RAM, running Linux (Ubuntu). The one SAN (10.10.12.3) is a 1.7GHZ laptop with 512 MB RAM also running Linux (Ubuntu).

C. Results

1) Discovery time: Figure 7 shows the results from the first scenario. This figure illustrates the time consumed by the processor when processing the information, since everything is done on one machine. The graphs clearly illustrate that context awareness takes longer time to process. The time difference is about 10-15ms, which we do not see as a serious time difference from a users perspective. Even with a strong machine as the SMN, there is a significantly increase in the response tim from 32 to 64 services, which is most likely caused by exhausting of computational resources. This seems also to be the case for the non context aware approach, however, here the increase is less. We consider that there is an upper limit of how many services can be running on one node on this machine, since each service started initiates a thread, which naturally leads many task and process switches at kernel level.

Figure 6 show the results from the local SAN-SMN discovery. The graphs in Figure 8 shows the result from the local SAN-SMN discovery. As expected context awareness adds to the discovery time. The added time is around 25ms, which

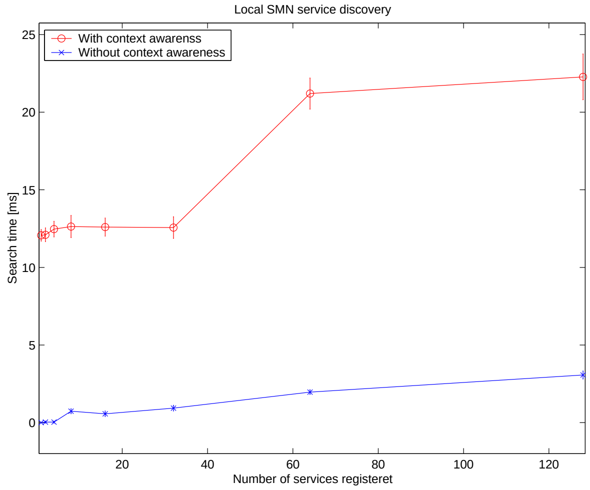

Fig. 7. Discovery time when doing SD locally on SMN with 95% confidence interval for each measured points

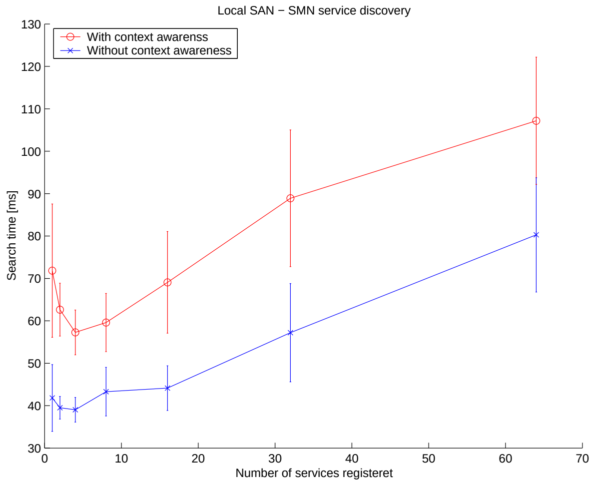

Fig. 8. Discovery time when doing SD locally from SAN to SMN with 95% confidence interval for each measurement point

complies with what was observed in the local test, plus some additional processing at the SAN.

The last result, seen in Figure 9, shows the discovery time from the global discovery done from a SAN. This test does not clearly show a difference between normal service discovery and CASD. This is most likely a matter of the INS overlay network, which seems to dominate the discovery time.

2) Traffic overhead: Table I provides an overview of the overhead generated during the tests. All numbers are averages evaluated from Etherreal. The effect of the ranking and selec-

	Packets	packets/s	bytes/packet	bytes/req	bytes/s
Local CASD	315	5,410	201	2120	1092
Local SD	388	6,421	263	3404	1689
Global CASD	917	7,6	272	8331	2058
Global SD	843	7,1	296	8309	2090

TABLE I

AVERAGE TRAFFIC OVERHEAD MEASURED AT THE SMN.

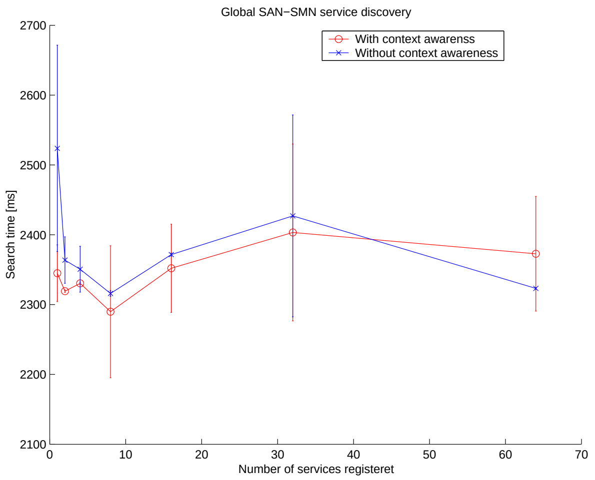

Fig. 9. Discovery time when doing global SD from the SAN with 95% confidence interval for each measurement point

tion process is clearly seen in the local part, as less data needs to be transferred to the SAN from the SMN. The picture is more blurred in the global discovery, since the SMN needs now to interact with both the SAN and the SMN overlay network, hereby making the deselection of services not that efficient in terms of network traffic.

V. Conclusion and outlook

In this paper we presented a system capable for performing CASD. The system is based on an experimental test bed that has been developed for Personal Networks in the MAGNET project[12]. In such a network, CASD will assist the user in the process of discovering and selecting services that are useful to him/her under a given situation. The system bases its decisions on a score value calculated by numerical context information for each service found. Making service discovery context-aware has some impact on performance. At the SMN the experimental tests showed an increases processing time at the order of 10-15ms, which is from a user's perspective not much. For the local SAN-SMN scenario, the measurements also showed an increased discovery time, however, in these settings the increase was in the area of 25ms. It was during the experiment observed less traffic was generated due the selection of services at the SMN, hereby leading to less service descriptions to be transmitted to the SAN. For global discovery, there was no clear difference between context and non CASD. This is mainly explained by that this overhead is negligible compared to what it takes for the overlay network to make discovery.

So far, we have looked at numerical context information for ranking services. In the future, we also want to look at non-numerical context information like favorite food or user activity. This requires a mapping of the context information to some numerical value. Another issue is how to deal with (partially) missing context information. Depending on the kind of context information, but possibly also user preferences, the service should be ignored or its score should be negatively influenced.

In addition to providing services reactively, i.e., as the result of a user request, we can also proactively search for services that may be of interest to the user in his/her current situation. This is especially useful for mobile users, as mobile devices have a display of limited size and manually entering information may be cumbersome. As a first step, the situation of the user has to be determined. Based on that, relevant services could be discovered and ranked, providing the user with direct access to these services.

VI. Acknowledgement

A thanks to the partners in the MAGNET project, especially those in WP2 for fruitful cooperation and discussion. A special thanks goes to Mark Girod Genet and Majid Ghader for their participation in the development of the MAGNET Service discovery framework.

About MAGNET Beyond:

MAGNET Beyond is a continuation of the MAGNET project (www.ist-magnet.org). MAGNET Beyond is a worldwide R&D project within Mobile and Wireless Systems and Platforms Beyond 3G. MAGNET Beyond will introduce new technologies, systems, and applications that are at the same time user-centric and secure. MAGNET Beyond will develop user-centric business model concepts for secure Personal Networks in multi-network, multi-device, and multi-user environments. MAGNET Beyond has 32 partners from 15 countries, among these highly influential Industrial Partners, Universities, Research Centres, and SMEs.

References

[1] I. Niemegeers and S. H. de Groot, "Personal networks: Ad hoc distributed personal environments," *Med-HocNet, IFIP Conference on Ad-Hoc Networks*, September 2002.

[2] P. Marina and et.al., "D2.1.2: Overall secure pn architecture," MAGNET Consortium - IST-507102, Tech. Rep., December 2005. [Online]. Available: http://www.ist-magnet.org

[3] "My personal adaptive global net and beyond." [Online]. Available: http://www.magnet.aau.dk/

[4] R.L.Olsen, H.Murakemi, H.P.Schwefel, and R.Prasad, "User centric service discovery in personal networks," *WPMC04*, September 2004.

[5] M. Ghader, R. L. Olsen, M. G. Genet, and R. Tafazolli, "Service management platform for personal networks," *Proceedings of IST Summit 2005, Dresden*, 2005.

[6] M. Khedr and A. Karmouch, "Enhancing service discovery with context information," *IEEE Canadian Conference of Electrical and Computer Engineering*, 2002.

[7] "Universal plug'n'play," 1999. [Online]. Available: http://www.upnp.org/

[8] *Service Location Protocol, svrloc - RFC2608*, V2 ed., IETF, June 1999.

[9] *Salutation Architecture Specification (Part-1)*, The Salutation Consortium, 1999. [Online]. Available: http://www.salutation.org/

[10] "Jini network technology." [Online]. Available: http://wwws.sun.com/software/jini/

[11] R. L. Olsen and et.al., "D2.2.3: Magnet service discovery specifications," MAGNET Consortium - IST-507102, Tech. Rep., December 2005.

[12] "My personal adaptive global net," 2004. [Online]. Available: http://www.ist-magnet.org/